

What is claimed is:

1. A method of tracking a network statistic stored within a collection of bits, comprising:

storing the collection of bits storing the network statistic as at least a first portion and a second portion, the first portion including a set of least-significant bits and the second portion including a set of more significant bits;

incrementing the first portion independently of the second portion based on a packet;

determining if the incrementing of the first portion caused a designated bit of the first portion to be set; and

if it is determined that the incrementing of the first portion caused the designated bit to be set:

incrementing the value stored by the second portion; and

resetting the designated bit within the first portion.

2. The method of claim 1, wherein incrementing the first portion comprises performing at least one memory operation on the first portion and performing no memory operations on the second portion.

3. The method of claim 1, wherein the network statistic comprises one of the following: a packet count and a byte count.

4. The method of claim 1, wherein the network statistic comprises one of the following: a per flow statistic, a per ingress interface statistic, and a per egress interface statistic.

5. The method of claim 1, further comprising:
reading the first portion; and
if it is determined that the designated bit of the first portion is not set,
reading the second portion.

6. The method of claim 1, further comprising:
determining a flow identifier of a packet based on packet characteristics;
and
based on the determined flow identifier, determining the location of at least one of the portions within a memory storing network statistics for multiple flows.

7. The method of claim 6, further comprising:
storing at least some of the first portions of network statistics for different flows in a first memory sub-system; and
storing second portions of the network statistics for the different flows in a second memory sub-system.

8. The method of claim 1, wherein

the network statistic comprises a value represented by 63-bits within a set of 64-bits;
the first portion comprises 32-bits; and
the second portion comprises 32-bits.

9. The method of claim 1, wherein the designated bit of the first portion comprises the most-significant bit of the first portion.

10. A method of tracking a network statistic stored within 64-bits, comprising:

storing an upper 32-bits of the 64-bits storing the network statistic in a first storage location, 63 of the 64-bits identifying the value of the network statistic;
storing a lower 32-bits of the 64-bit value in a second storage location;
incrementing the value stored by the lower 32-bits independently of the upper 32-bits based on a received packet; and
in response to the setting of the most significant bit of the lower 32-bits:
incrementing the upper 32-bits; and
resetting the most significant bit of the lower 32-bits.

11. The method of claim 10, wherein the network statistic comprises one of the following: a packet count and a byte count.

12. The method of claim 10, wherein the network statistic comprises one of the following: a per flow statistic, a per ingress interface statistic, and a per egress interface statistic.

13. The method of claim 10, further comprising:
reading the lower 32-bits of the 64-bit value; and
if it is determined that the most significant bit of the lower 32-bits is not set,
reading the upper 32-bits of the 64-bit value.

14. The method of claim 10, further comprising
determining a flow identifier of a packet based on packet characteristics;
and

based on the determined flow identifier, determining the location of one of the 32-bit portions within a table storing statistics for multiple flows.

15. The method of claim 14, further comprising:
storing at least some of the upper 32-bit portions of network statistics for multiple flows in a first memory sub-system; and
storing lower 32-bit portions of network statistics for multiple flows in a second memory sub-system.

16. A computer program product, disposed on a computer readable medium, to track a network statistic stored within a collection of bits, the program including instructions for causing at least one processor to:

store the bits storing the network statistic as at least a first portion and a second portion, the first portion including a set of least-significant bits and the second portion including a set of more significant bits of the bits of the network statistic;

increment the first portion independently of the second portion based on a received packet;

determine if the incrementing of the first portion caused a designated bit of the first portion to be set; and

if it is determined that the incrementing of the first portion caused the designated bit to be set:

increment the value stored by the second portion; and
reset the designated bit within the first portion.

17. The program of claim 16, wherein instructions for causing the at least one processor to increment the first portion comprise instructions for causing the at least one processor to perform at least one memory operation on the first portion and perform no memory operations on the second portion.

18. The program of claim 16, wherein the network statistic comprises one of the following: a packet count and a byte count.

19. The program of claim 16, wherein the network statistic comprises one of the following: a per flow statistic, a per ingress interface statistic, and a per egress interface statistic.

20. The program of claim 16, further comprising instructions for causing at least one processor to:

read the first portion; and
if it is determined that the most significant bit of the first portion is not set,
read the second portion.

21. The program of claim 16, further comprising instructions for causing at least one processor to:

determine a flow identifier of a packet based on packet characteristics;
and
based on the determined flow identifier, determine the location of at least one of the portions within a memory storing network statistics for multiple flows.

22. The program of claim 21, further comprising instructions for causing at least one processor to:

store first portions of network statistics for multiple flows in a first memory sub-system; and

store second portions of the network statistics for the multiple flows in a second memory sub-system.

23. The program of claim 16, wherein
the network statistic value comprises 63-bits included within a set of 64-bits;

the first portion comprises 32-bits; and
the second portion comprises 32-bits.

24. The program of claim 16, further comprising instructions to cache the first portion in a local memory of a network processor packet engine.

25. A network processor, comprising:
at least one interface to receive data of packets received over a network;
multiple packet engines; and
a computer program product, disposed on a computer readable medium,
to cause the multiple packet engines to:
store the bits storing the network statistic as at least a first portion
and a second portion, the first portion including a set of least-significant bits and
the second portion including a set of more significant bits of the bits of the
network statistic;
increment the first portion based on a received packet;

determine if the incrementing of the first portion caused a designated bit of the first portion to be set; and

if it is determined that the incrementing of the first portion caused the designated bit to be set:

increment the value stored by the second portion; and

reset the designated bit within the first portion.

26. The processor of claim 25, wherein instructions for causing at least one engine to increment the first portion comprise instructions for causing the at least one engine to perform at least one memory operation on the first portion and perform no memory operations on the second portion.

27. The processor of claim 25, wherein the network statistic comprises one of the following: a packet count and a byte count.

28. The processor of claim 25, further comprising instructions for causing at least one engine to:

read the first portion; and

if it is determined that the designated bit of the first portion is not set, read the second portion.

29. The processor of claim 25, further comprising instructions for causing at least one engine to:

determine a flow identifier of a packet based on packet characteristics;

and

based on the determined flow identifier, determine the location of at least one of the portions within a memory storing network statistics for multiple flows.

30. The processor of claim 29, further comprising instructions for causing at least one of the engines to:

store first portions of network statistics for multiple flows in a first memory sub-system; and

store second portions of network statistics for multiple flows in a second memory sub-system.

31. The network processor of claim 25, wherein the multiple packet engines comprise engines having hardware contexts for different execution threads.

32. A system to process packets received over a network, the system comprising:

multiple line cards, an individual line card including:

at least one physical layer component (PHY); and

at least one network processor having multiple packet engines having access to instructions to cause the at least one of the multiple packet engines to:

store the bits storing the network statistic as at least a first portion and a second portion, the first portion including a set of least-significant bits and the second portion including a set of more significant bits of the bits of the network statistic;

increment the first portion based on a received packet;

determine if the incrementing of the first portion caused a most significant bit of the least-significant bits to be set; and

if it is determined that the incrementing of the least-significant bits caused the most significant bit of the first portion to be set:

increment the value stored by the more significant bits; and

reset the most significant bit of the first portion; and
a switch fabric interconnecting the multiple line cards.

33. The system of claim 32,

wherein the network statistic comprises one of the following: a packet count and a byte count;

wherein the network statistic comprises one of the following: a per flow statistic, a per ingress interface statistic, and a per egress interface statistic;

wherein the network statistic comprises a value identified by 63-bits included within a set of 64-bits;

wherein the first portion comprises 32-bits;

wherein the second portion comprises 32-bits;

further comprising instructions for causing at least one engine to:

determine a flow identifier of a packet based on packet characteristics; and

based on the determined flow identifier, determine the location of at least one of the portions within a memory storing network statistics for multiple flows; and

further comprising instructions for causing at least one processor to:

read the first portion; and

if it is determined that the most significant bit of the first portion is set, read the second portion.